Dealing with complicating factors in practice: focus on confounding

- **Classical PCA**
- **PCA in GWAS -GenABEL**
- **PRsice**

Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components

Computing the Principal Components (PC)

- **Lets use classical iris dataset.**
- **The data contain four continuous variables which corresponds to physical measures of flowers and a categorical variable describing the flowers' species.**

```
# Load data
data(iris)
head(iris, 3)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
```

Computing the Principal Components (PC)

- **We will apply PCA to the four continuous variables and use the categorical variable to visualize the PCs later.**
- **In the following code we apply a log transformation to the continuous variables and set center and scale. equal to TRUE in the call to prcomp to standardize the variables prior to the application of PCA:**

```
# log transform
log.ir <- log(iris[, 1:4])
ir.species <- iris[, 5]

# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
ir.pca <- prcomp(log.ir,
                 center = TRUE,
                 scale. = TRUE)
```

Principal Components (PC)

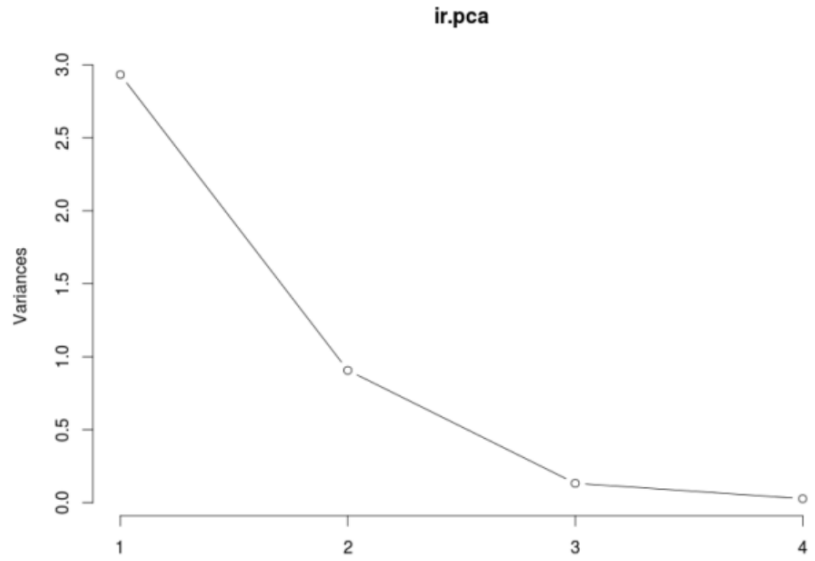
- **The prcomp function returns an object of class prcomp, which have some methods available.**
- **The print method returns the standard deviation of each of the four PCs, and their rotation (or loadings), which are the coefficients of the linear combinations of the continuous variables.**

```
# print method
print(ir.pca)

Standard deviations:
[1] 1.7124583 0.9523797 0.3647029 0.1656840

Rotation:
                    PC1         PC2        PC3         PC4
Sepal.Length  0.5038236 -0.45499872  0.7088547  0.19147575
Sepal.Width  -0.3023682 -0.88914419 -0.3311628 -0.09125405
Petal.Length  0.5767881 -0.03378802 -0.2192793 -0.78618732
Petal.Width   0.5674952 -0.03545628 -0.5829003  0.58044745
```

Principal Components (PC)

- **The plot method returns a plot of the variances (y-axis) associated with the PCs (x-axis).**
- **The Figure below is useful to decide how many PCs to retain for further analysis. In this simple case with only 4 PCs this is not a hard task and we can see that the first two PCs explain most of the variability in the data.**

```
plot(ir.pca, type = "l")
```



Result of Principal Components (PC)

- The summary method describe the importance of the PCs.
- The first row describe again the standard deviation associated with each PC.
- The second row shows the proportion of the variance in the data explained by each component while the third row describe the cumulative proportion of explained variance.
- We can see there that the first two PCs accounts for more than of the variance of the data.

```
# summary method
summary(ir.pca)

Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7125 0.9524 0.36470 0.16568
Proportion of Variance 0.7331 0.2268 0.03325 0.00686
Cumulative Proportion  0.7331 0.9599 0.99314 1.00000
```

Plotting Principal Components (PC)

Download following packages

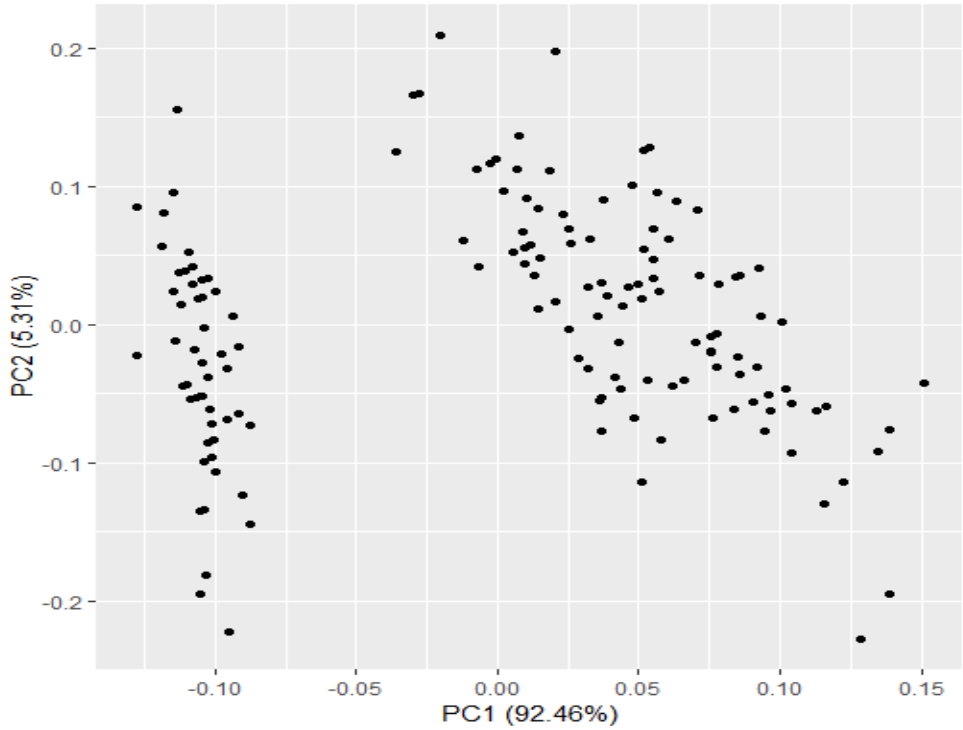
https://cran.r-project.org/web/packages/ggplot2/index.html

https://cran.r-project.org/web/packages/ggfortify/index.html

Ready for PCA plotting !!!!

Plotting PCA (Principal Component Analysis)

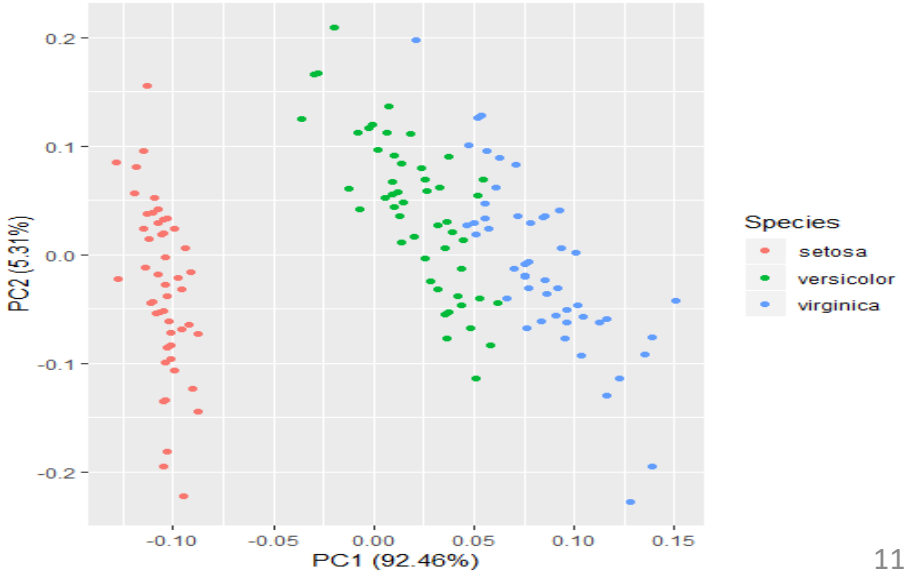
```
> library(ggfortify)
> df <- iris[c(1, 2, 3, 4)]
> autoplot(prcomp(df))
```



Plotting PCA (Principal Component Analysis)

- **PCA result should only contains numeric values.**
- **If you want to colorize by non-numeric values which original data has, pass original data using data keyword and then specify column name by colour keyword.**
- **Use help(autoplot.prcomp) (or help(autoplot.*) for any other objects) to check available options.**

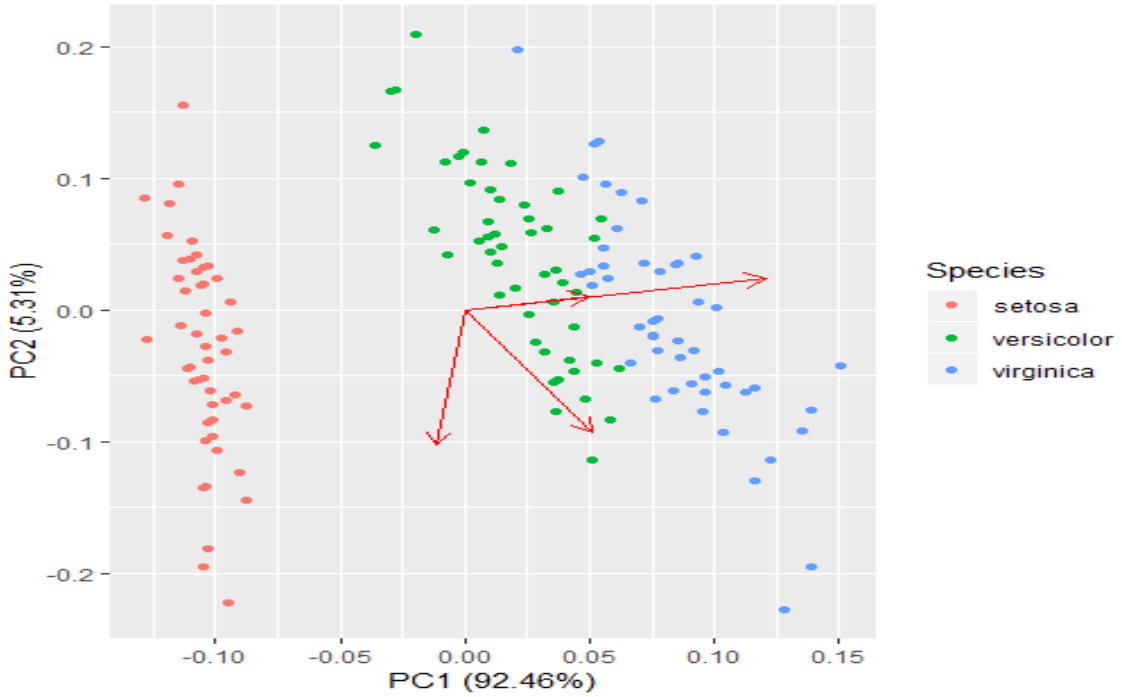
```
autoplot(prcomp(df), data = iris, colour = 'Species')
```



Eigenvectors in PCA (Principal Component Analysis)

- Passing loadings = TRUE draws eigenvectors.

autoplot(prcomp(df), data = iris, colour = 'Species', loadings = TRUE)



PCA (Principal Component Analysis) in R

- prcomp(x, retx = TRUE, center = TRUE, scale. = FALSE, tol = NULL, ...)
- princomp(formula, data = NULL, subset, na.action, ...)

PCA (Principal Component Analysis) in GWAS

Why are PCA used?

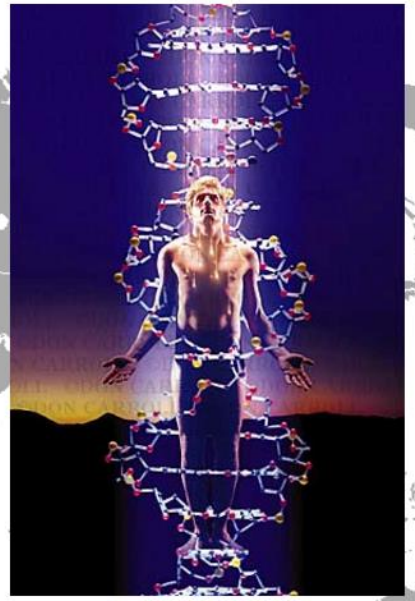
- **In GWAS context, PCA is mainly used to account for population-specific variations in alleles distribution on the SNPs (with the SNP case) under investigation.**

- **Such "population substructure" mainly arises as a consequence of varying frequencies of minor alleles in genetically distant ancestries (e.g. japanese and black-african or european-american).**


Diversity
• Human
• Plants
• Animals
• Bacteria
• etc

Human Diversity

Medicine and Treatment



- **The general idea is well explained in Population Structure and Eigenanalysis, by Patterson et al. (*PLoS Genetics* 2006, 2(12))**

OPEN ACCESS Freely available online PLoS GENETICS

Population Structure and Eigenanalysis

Nick Patterson[1*], Alkes L. Price[1,2], David Reich[1,2]

1 Broad Institute of Harvard and MIT, Cambridge, Massachusetts, United States of America, 2 Department of Genetics, Harvard Medical School, Boston, Massachusetts, United States of America

Current methods for inferring population structure from genetic data do not provide formal significance tests for population differentiation. We discuss an approach to studying population structure (principal components analysis) that was first applied to genetic data by Cavalli-Sforza and colleagues. We place the method on a solid statistical footing, using results from modern statistics to develop formal significance tests. We also uncover a general "phase change" phenomenon about the ability to detect structure in genetic data, which emerges from the statistical theory we use, and has an important implication for the ability to discover structure in genetic data: for a fixed but large dataset size, divergence between two populations (as measured, for example, by a statistic like F_{ST}) below a threshold is essentially undetectable, but a little above threshold, detection will be easy. This means that we can predict the dataset size needed to detect structure.

Citation: Patterson N, Price AL, Reich D (2006) Population structure and eigenanalysis. PLoS Genet 2(12): e190. doi:10.1371/journal.pgen.0020190

Introduction

A central challenge in analyzing any genetic dataset is to explore whether there is any evidence that the samples in the data are from a population that is structured. Are the individuals from a homogeneous population or from a population containing subgroups that are genetically distinct? Can we find evidence for substructure in the data, and practical, even on the largest datasets. This is our main aim in this paper.

Using some recent results in theoretical statistics, we introduce a formal test statistic for population structure. We also discuss testing for *additional* structure after some structure has been found. Finally, we are able to estimate the degree of population differentiation that will be detectable for a given data size.

How are PCA calculated in GWAS

- **The construction of principal axes follows from the classical approach to PCA, which is applied to the scaled matrix (individuals by SNPs) of observed genotypes (AA, AB, BB; say B is the minor allele in all cases), to the exception that an additional normalization to account for population drift might be applied.**

- **It all assumes that the frequency of the minor allele (taking value in {0,1,2}) can be considered as numeric, that is we work under an *additive model* (also called allelic dosage) or any equivalent one that would make sense.**

PCA for SNPs

- X is the M x N matrix, where M is a number of individuals and N is a number of SNPs.

$$XX^T = UDV^T$$

U is the matrix of eigenvectors or PC scores.

$$B^T = D^{-1/2}U^TX$$

B is the factor loadings

$$PCs = X.B$$

Normalization for SNPs

- Zero means

If X is a vector

$$M = X - \text{mean}(X)$$

- Unit variance

$$Y = M / \text{sd}(X)$$

- In R, it is more efficient to use apply() with mean() and sd()

Which tools to use to do PCA for GWAS?

- **In previous practical session , we performed QQ plot Association testing and Manhatton plotting.**

- **Today's session, we will focus on PCA for detail interpretation with use of GenABEL package**

GenABEL

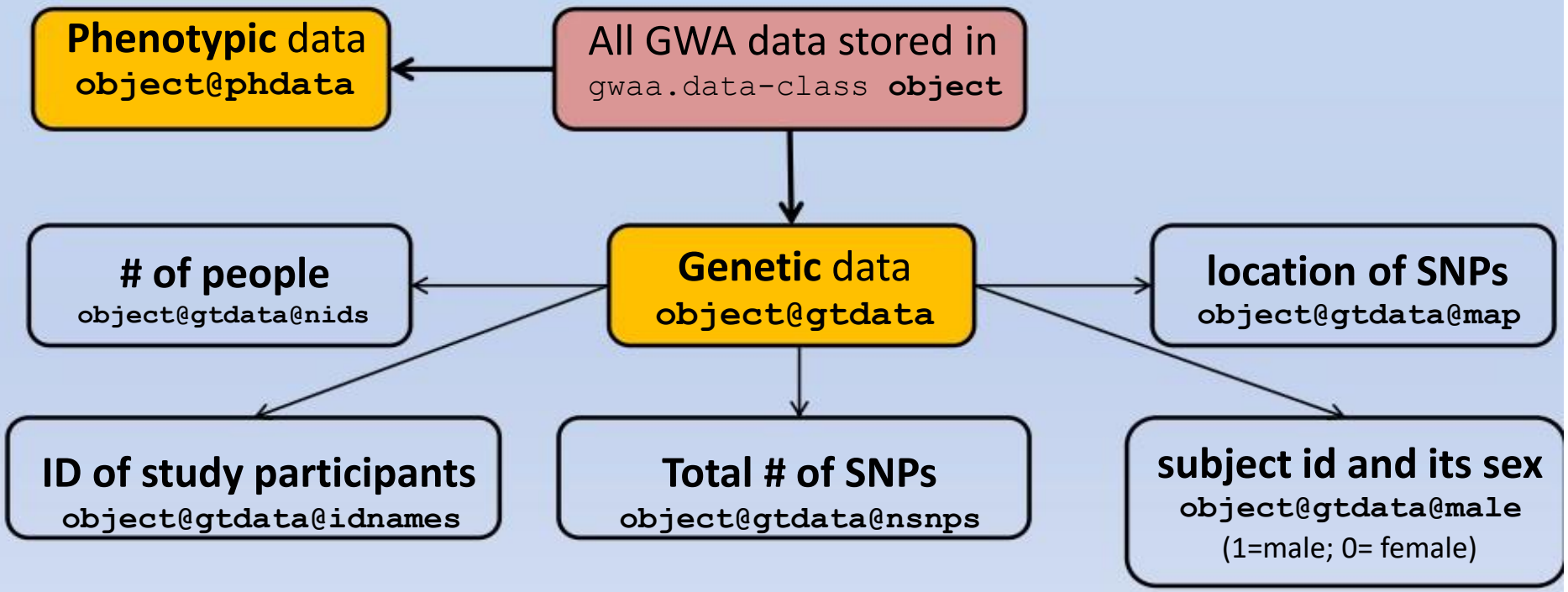
- number of SNPs: gwaa_object@gtdata@nsnps

- SNP names: gwaa_object@gtdata@snpnames

- Chromosome labels:gwaa_object@gtdata@chromosome

- SNPs map positions: gwaa_object@gtdata@map

GeneABEL object structure



PCA in GenABEL

```
# Lets run dataset
library(GenABEL)
data(ge03d2ex)
#Perform the QC check as performed in previous session
QCresults <- check.marker(ge03d2ex,p.level=0)
#Perform PCA on filtered Quality data
Cdata <- ge03d2ex[QCresults$idok, QCresults$snpok]
```

Finding population structure (1)

- Need to detect individuals that are “genetic outliers” compared to the rest using SNP data

  - compute matrix of genetic kinship between subjects of this study

```
Cdata.gkin <- ibs(Cdata[,autosomal(Cdata)],weight="freq")
```

```
Cdata.gkin[1:5,1:5]
```

	id199	id300	id403	id415	id666
id199	0.494427766	3255.00000000	3253.00000000	3241.00000000	3257.0000000
id300	-0.011754266	0.49360296	3261.00000000	3250.00000000	3264.0000000
id403	-0.012253378	-0.01262949	0.50541775	3247.00000000	3262.0000000
id415	-0.001812109	0.01388179	-0.02515438	0.53008236	3251.0000000
id666	-0.018745051	-0.02127344	0.02083723	-0.02014175	0.5306584

- The numbers below the diagonal show the genomic estimate of kinship ('genome-wide IBD'),
- The numbers on the diagonal correspond to 0.5 plus the genomic homozigosity
- The numbers above the diagonal tell how many SNPs were typed successfully for both subjects

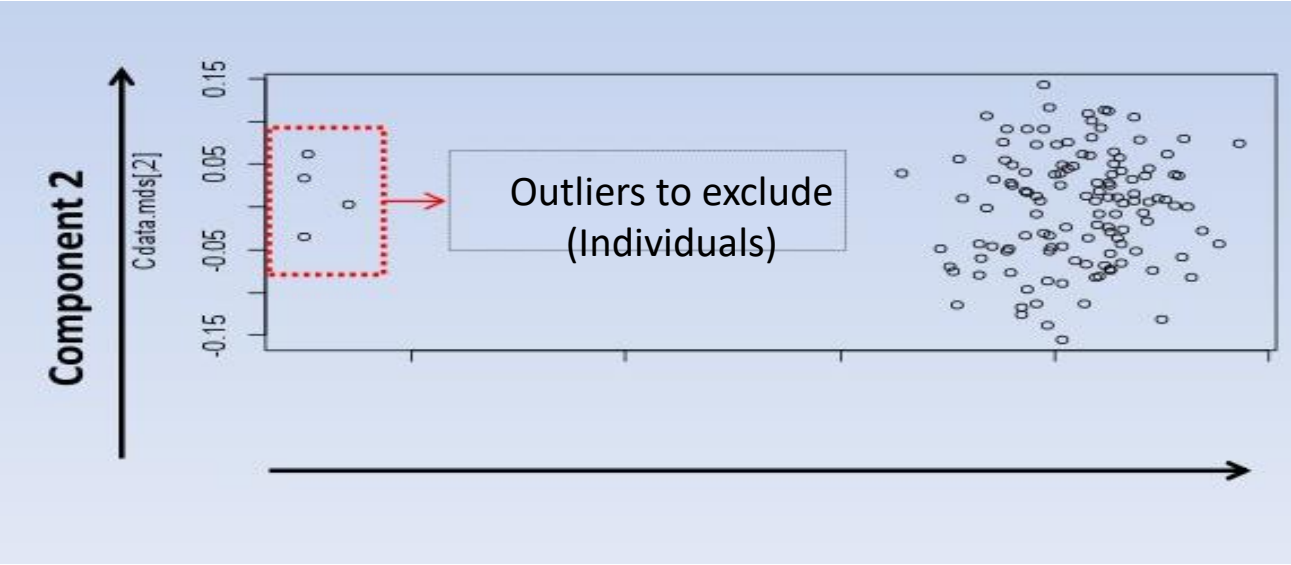
Finding population structure (2)

2. Compute distance matrix from previous

```
Cdata.dist <- as.dist(0.5-Cdata.gkin)
```

3. Do Classical Multidimensional Scaling (PCA) and visualize results

```
Cdata.mds <- cmdscale(Cdata.dist)
plot(Cdata.mds)
```



- **The PCA fitted the genetic distances along the 2 components**
- **Points are individuals**
- **There are clearly two clusters**
- **Need to select all individuals from biggest cluster**

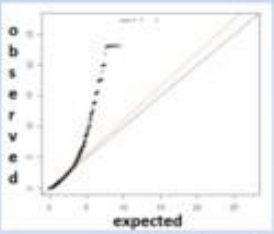
What is next after GWAS association ?

Large cohort (>1000) of cases and controls

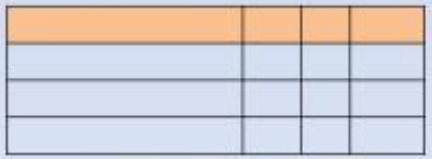
Get genome information with SNP arrays

Find deviating from expected haplotypes
visualize SNP-SNP interactions using HapMap

Detection of potential association signals and their fine mapping (e.g. detection of LD, stratification effect)

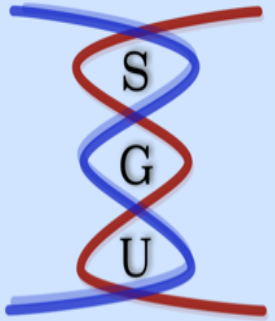


Replication of detected association in new cohot / subset for validation purposes



PRSice: Polygenic Risk Score software

by Jack Euesden, Cathryn Lewis & Paul O'Reilly



Statistical Genetics Unit
King's College London

PRSice-2 out now!!

PRSice has been recoded into C++ by Shing Wan (Sam) Choi, making it much faster, suitable for Biobank-sized data (no intermediary files), and has several new features, such as bgen input, empirical P-value output and PRSet: PRS calculated across pathways / gene sets. Get the new version of PRSice from the link below (the rest of this webpage still relates to the original version of PRSice in case you still want to use that, but soon we will move entirely to PRSice-2).

Features of PRSice

- **High-resolution scoring (PRS calculated across a large number of P-value thresholds)**
- **Identify Most predictive PRS**
- **Empirical P-values output (not subject to over-fitting)**
- **Genotyped (PLINK binary) and imputed (Oxford bgen v1.2) data input**
- **Biobank-scale genotyped data can be analysed within hours**

Features of PRSice

- **Application across multiple target traits simultaneously**
- **Results plotted in several formats (bar plots, high-res plots, quantile plots)**
- **PRSet: function for calculating PRS across user-defined pathways / gene sets**
- **Incorporation of covariates**

PRSice Processing : Steps

- PRSice ('precise') implements a pipeline
- GWAS results are obtained on a phenotype of interest (we call the base phenotype)
- A target data set is obtained, which contains genotype and phenotype data (the target phenotype may be the same or different to the base phenotype).

SNP	CHR	BP	A1	A2	P	OR
SNP_22857	4	103593179	1	2	0.2852	13.29
SNP_13879	2	237416793	1	2	0.8784	21.624
SNP_20771	4	16957461	1	2	0.1994	91.265
SNP_13787	2	235355721	1	2	0.7234	3.178
SNP_25383	4	189927377	1	2	0.3309	3.167
SNP_25290	4	187995996	1	2	0.6327	0.427
SNP_21478	4	40161304	1	2	0.06454	5.066
SNP_12129	2	176643771	1	2	0.9378	1.276
SNP_22809	4	101441465	1	2	0.8111	0.004

- **Polygenic Risk Scores (PRS) are calculated for individuals in the target data as a sum of 'risk alleles' across SNPs with (GWAS) P-values below a given P-value threshold, weighted by the effect sizes estimated by the GWAS (the scores thus correspond to a genetic risk/burden of the base phenotype for each individual).**

- **Model fit and significance illustrated via bar plots, scatter plots and quantile plots.**

- **A regression is performed to test the association between the PRS and the target phenotype (eg. to test for shared genetic aetiology). Covariates, as well as ancestry informative variables for controlling for population structure (which can be calculated in PRSice), can be included in this regression.**

- **PRS and the subsequent regressions are calculated at a number of P-value thresholds and the model fit of the regressions assessed. PRSice repeats the analysis at 1000s of thresholds in order to identify the most predictive threshold and model.**

Polygenic risk score analysis with PRSice

- When PLINK genotype target files are available, PRSice provides a relatively easy way of performing polygenic risk score analysis.
- Instead of performing a polygenic risk score analysis on all genetic variants it is customary to clump first.
- In clumping, within each block of correlated SNPs the SNP with the lowest p-value in the discovery set is selected and all other SNPs are ignored in downstream analyses.

Installation of PRSice

- **PRSice is an R program that can be run in Unix, Linux and Mac OS from the command line.**

- **First download and unpack the PRSice software with the following code on the command line.**

```
wget http://prsice.info/PRSice_v1.25.zip
unzip PRSice_v1.25.zip
```

Installation of PRSice

- **If you enter the newly created PRSice_v1.25 directory you see the PRSice R script (PRSice_v1.25.R), binary files (for linux and mac) of plink1.9 (also known als plink2), user documentation, and sample datasets to perform a polygenic risk analysis.**

cd PRSice_v1.25
ls

Dependency of PRSice

In addition, PRSice requires the following R packages to be preinstalled:

- **ggplot2**
- **plyr**
- **batch**
- **fmsb**
- **gtx**

Ready to use PRSice !!!!

Running PRSice

- To run a polygenic risk score analysis on the toy data provided with PRSice run the following code, which is tested in R versions 3.2.1 and 3.2.2.

```
> Rscript PRSice_v1.25.R -q --args plink ./plink_1.9_linux_160914 base TOY_BASE_GWAS.assoc  target TOY_TARGET_DATA slower 0 sinc 0.01 supper 0.5
```

Running PRSice – detail description

- The *plink* parameter specifies where PRSice can find the correct plink executable with which to perform the analysis. Change this to the Mac executable if appropriate (./plink_1.9_mac_160914).
- The *base* parameter refers to the file with summary statistics from the *base* sample (also known as discovery or training samples). These summary statistics contain for each genetic variant at least an effect size and p-value.
- The *target* parameter refers to the prefix of the files (without file extension) that contain the genotype data in binary plink format (i.e.; .bed,.bim,.fam file extensions)

Running PRSice – detail description

- Sample overlap across the discovery and target sample will greatly inflate the association between the polygenic risk score and the disease trait.
- The last three parameters indicate that several polygenic risk score analyses are performed with p-value thresholds from 0 to 0.5 with steps of 0.1. That is, in each risk score analysis only SNPs with a below a particular threshold in the base sample are used in the prediction model.

Interpreting the results : PRS model-fit

- A file containing the PRS model fit across thresholds is named PRSice RAW RESULTS DATA.txt;
- This is stored as threshold, P-value, variance in target phenotype explained, r2, and number of SNPs at this threshold.

thresh	p.out	r2.out	nsnps
0.01	0.830154073713849	0.00003067529440429 58	534
0.02	0.47183826387497	0.00034519510578 1867	1216
0.03	0.00000193633357794 93	0.01528215621463 69	1941
0.04	0.00000448497018985 22	0.01416881204516 33	2684

Interpreting the results : Scores for each individual

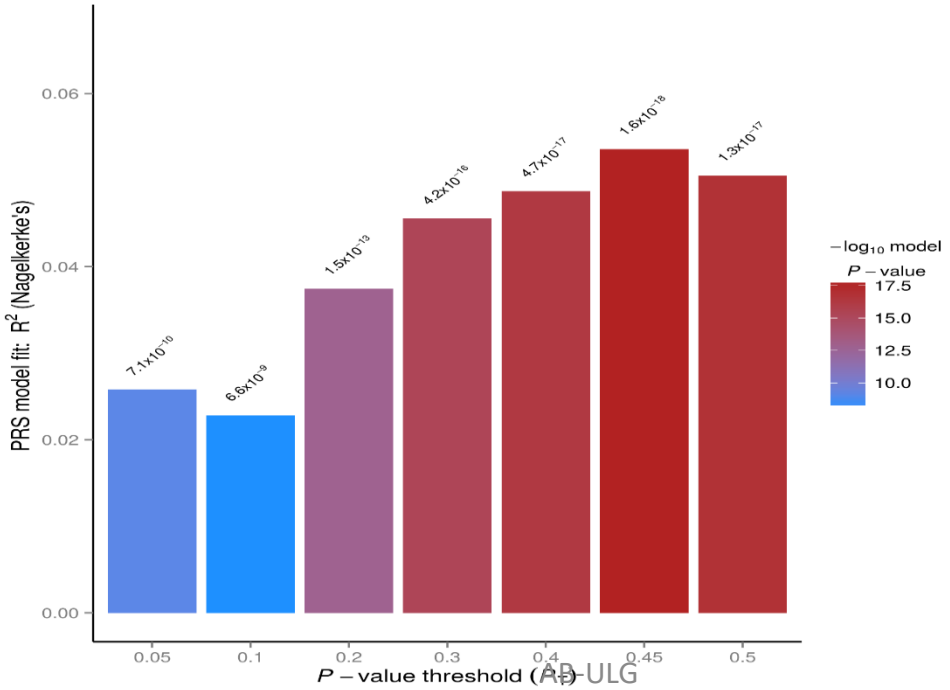
- **A file containing PRS for each individual at the best-fit PRS named PRSice SCORES AT BEST-FIT-PRS.txt (by default)**

```
IID pT_0.45
CAS_1 -0.00661597
CAS_10 0.00171366
CAS_100 -0.00599495
CAS_1000 0.00262721
CAS_101 -0.005297
CAS_102 -0.0011709
```

Interpreting the results

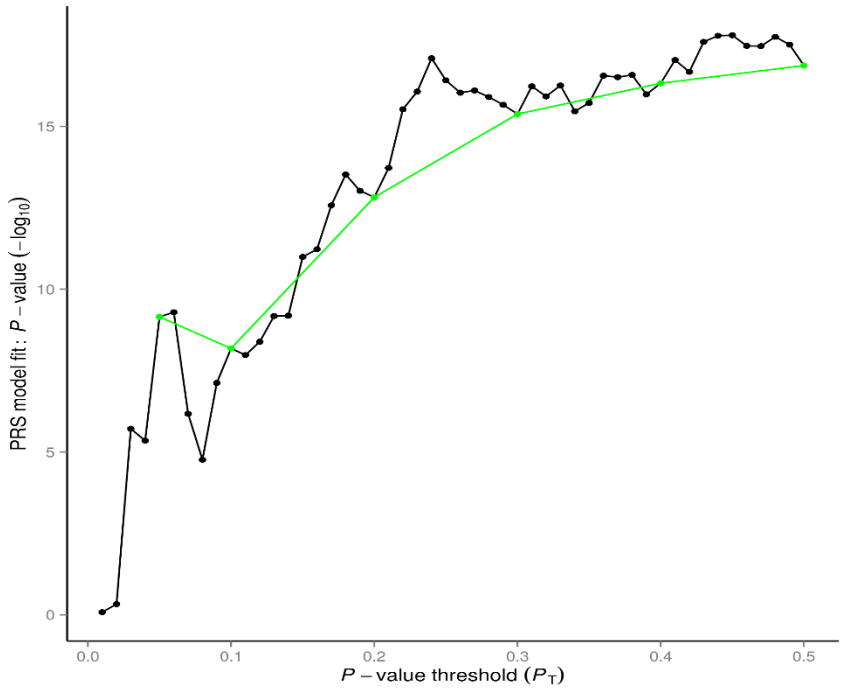
- PRSice saves two plots and and several text files.

- The first plot is PRSice_BARPLOT. This plot shows the predictive value (Nagelkerke's) in the target sample of models based on SNPs with p-values below specific thresholds in the base sample.



Interpreting the results

- **The second plot is PRSice_HIGH-RES_PLOT (Figure S2) shows for many different p-value thresholds the p-value of the predictive effect ( ) in black together with an aggregated trend line in green.**



References:

[1] Venables, W. N., Brian D. R. Modern applied statistics with S-PLUS. Springer-verlag. (Section 11.1)

[2] Box, G. and Cox, D. (1964). An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological) 211-252